

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for dynamically selecting functionally equivalent Web services through a single autonomic proxy, comprising:
 - receiving a client request to locate a Web service at the single autonomic proxy;
 - querying a policy discovery mechanism based on the client request;
 - ~~locating~~ using the autonomic proxy to locate multiple Web services candidates to service ~~serve~~ the client request, wherein each Web service candidate is functionally equivalent to the other Web service candidates[[:]], and the Web service candidates collectively comprise a group;
 - ~~determining which Web service candidate to invoke based on the Web service candidate business policy;~~
 - using the single autonomic proxy to select a first Web service to invoke from the group of Web service candidates, wherein the selection is based on the business policy of the first Web service;
 - sending a message to the first Web service from the autonomic proxy to determine if the first Web service is available;
 - in response to a determination by the autonomic proxy that the first Web service is not available, operating the dynamic proxy to discover the business policy of each of the other service candidate in the group of Web service candidates;
 - in response to a determination that the first Web service is not available, operating the autonomic proxy to dynamically select a second Web service from the group of Web service candidates based on the business policy; and
 - sending a request to the second Web service to serve the client request.
2. (Currently Amended) The method of claim 1, wherein the ~~policy discovery mechanism is UDDI.~~ autonomic proxy is disposed to measure the response times of each Web service by sending messages to each of the Web service candidates.
3. (Currently Amended) The method of claim [[1]] 2, wherein the ~~Web service is described using WSDL.~~ autonomic proxy dynamically selects the Web service that is responding most quickly according to its business policy to be the first Web service.

4. (Currently Amended) The method of claim [[3]] 1, wherein the Web service is described using WSDL, and querying the policy discovery mechanism includes obtaining a WSDL Web service interface description for the requested Web service.
5. (Currently Amended) The method of claim [[3]] 4, wherein querying the policy discovery mechanism includes locating a wsdlSpec tModel based on the WSDL Web service interface description for the requested Web service.
6. (Original) The method of claim 1, wherein determining which Web service candidate to invoke based on the Web service candidate business policy includes analyzing business criteria of the Web service candidate.
7. (Original) The method of claim 6, wherein the business criteria includes cost of service.
8. (Canceled)
9. (Original) The method of claim 1, further comprising:
analyzing a metadata about the client request.
10. (Original) The method of claim 9, wherein the metadata includes Web service response time information.
11. (Currently Amended) The method of claim 1, wherein the locating step includes
discovering the business policy of each Web service candidate in the group of Web service candidates;
dynamically selecting the Web service from the group of Web service candidates responding the quickest based on the business policy; and
sending a request to the selected Web service to ~~service~~ serve the client request.
12. (Original) The method of claim 1, wherein the business policy includes Web Services Policy Framework (WSPolicy).

13. (Currently Amended) A data processing system for dynamically selecting functionally equivalent Web services through a single autonomic proxy, comprising:

receiving means for receiving a client request to locate a Web service at the autonomic proxy;

querying means for querying a policy discovery mechanism based on the client request;

locating means for ~~locating~~ operating the autonomic proxy to locate multiple Web service candidates to ~~serve~~ serve the client request, wherein each Web service candidate is functionally equivalent to the other Web service candidates [[; and]], and the Web service candidates collectively comprise a group;

~~determining means for determining which Web service candidate to invoke based on the Web service candidate business policy;~~

first selecting means for using the single autonomic proxy to select a first Web service to invoke from the group of Web service candidates, wherein the selection is based on the business policy of the first Web service;

first sending means for sending a message to the first Web service from the autonomic proxy to determine if the first Web service is available;

discovering means responsive to a determination by the autonomic proxy that the first Web service is not available, for operating the dynamic proxy to discover the business policy of each of the other service candidates in the group of Web service candidates;

second selecting means, responsive to a determination that the first Web service is not available for operating the autonomic proxy to dynamically select a second Web service from the group of Web service candidates based on the business policy; and

sending means for sending a request to the second Web service to serve the client request.

14. (Original) The data processing system of claim 13, wherein the policy discovery mechanism is UDDI.

15. (Original) The data processing system of claim 13, wherein the Web service is described using WSDL.

16. (Original) The data processing system of claim 15, wherein the querying means includes obtaining a WSDL Web service interface description for the requested Web service.

17. (Original) The data processing system of claim 15, wherein querying means includes locating a wsdlSpec tModel based on the WSDL Web service interface description for the requested Web service.

18. (Original) The data processing system of claim 13, wherein the determining means includes analyzing business criteria of the Web service candidate.
19. (Original) The data processing system of claim 18, wherein the business criteria includes cost of service.
20. (Canceled)
21. (Original) The data processing system of claim 13, further comprising:
analyzing means for analyzing a metadata about the client request.
22. (Original) The data processing system of claim 21, wherein the metadata includes Web service response time information.
23. (Currently Amended) The data processing system of claim 13, wherein the locating means includes
discovering means for discovering the business policy of each Web service candidate in the group of Web service candidates;
selecting means for dynamically selecting the Web service from the group of Web service candidates responding the quickest based on the business policy; and
sending means for sending a request to the selected Web service to ~~service~~ serve the client request.
24. (Original) The data processing system of claim 11, wherein the business policy includes Web Services Policy Framework (WSPolicy).
25. (Currently Amended) A computer program product in [[a]] an executable computer readable medium for dynamically selecting functionally equivalent Web services through a single autonomic proxy, comprising:
first instructions for receiving a client request to locate a Web service at the autonomic proxy;
second instructions for querying a policy discovery mechanism based on the client request;
third instructions for ~~locating~~ operating the autonomic proxy to locate multiple Web services candidates to ~~service~~ serve the client request, wherein each Web service candidate is functionally

equivalent to the other Web service candidates[[]; and]], and the Web service candidates collectively comprise a group;

~~fourth instructions for determining which Web service candidate to invoke based on the Web service candidate business policy;~~

fourth instructions for using the single autonomic proxy to select a first Web service to invoke from the group of Web service candidates, wherein the selection is based on the business policy of the first Web service;

fifth instructions for sending a message to the first Web service from the autonomic proxy to determine if the first Web service is available;

sixth instructions for operating the dynamic proxy to discover the business policy of each of the other service candidates in the group of Web service candidates, in response to a determination by the autonomic proxy that the first Web service is not available;

seventh instructions for operating the autonomic proxy to dynamically select a second Web service from the group of Web service candidates based on the business policy, in response to a determination that the first Web service is not available; and

eighth instructions for sending a request to the second Web service to serve the client request.

26. (Original) The computer program product of claim 25, wherein the policy discovery mechanism is UDDI.

27. (Original) The computer program product of claim 25, wherein the Web service is described using WSDL.

28. (Original) The computer program product of claim 27, wherein the querying instructions include obtaining a WSDL Web service interface description for the requested Web service.

29. (Original) The computer program product of claim 25, wherein the querying instructions include locating a wsdlSpec tModel based on the WSDL Web service interface description for the requested Web service.

30. (Original) The computer program product of claim 25, wherein the determining instructions include analyzing business criteria of the Web service candidate

31. (Original) The computer program product of claim 30, wherein the business criteria includes cost of service.
32. (Canceled)
33. (Original) The computer program product of claim 25, further comprising:
fifth instructions for analyzing a metadata about the client request.
34. (Original) The computer program product of claim 33, wherein the metadata includes Web service response time information.
35. (Currently Amended) The computer program product of claim 25, wherein the locating instructions include
instructions for discovering the business policy of each Web service candidate in the group of Web service candidates;
instructions for dynamically selecting the Web service from the group of Web service candidates responding the quickest based on the business policy; and
instructions for sending a request to the selected Web service to ~~service~~ serve the client request.
36. (Original) The computer program product of claim 25, wherein the business policy includes Web Services Policy Framework (WSPolicy).